

CSCD01 Deliverable 4

Team 34: 3+3

User Guide

For Deliverable 4 we have chosen to fix 4 major bugs in Mozilla's [donate-wagtail](#) project, as discussed in the previous deliverable.

Issue #661: "Other amount" text field allows amounts in scientific notation

Changes can be found in the

<https://github.com/CSCD01/donate-wagtail-team34/tree/issue-661> branch. To see all added code, see the change comparison here:

<https://github.com/mozilla/donate-wagtail/compare/master...CSCD01:issue-661>

This bug fix prevents the user inputting numbers in scientific notation in the custom donation amount field on the donate.mozilla.org webpage. Previously, if the user were to enter the string '1e3' in the custom amount field, the website would attempt to make a donation of 1000 dollars, which is a problem if the user accidentally presses the 'e' key when trying to donate 13 dollars, for example.

The bug fix prevents users from inputting any non-numeric characters (asides from the decimal point) by clearing the text input when a non-numeric character is inputted on the front-end, and further sanitizing the input on the back-end by defaulting to a donation of \$0 when donating amounts in scientific notation.

Amounts in scientific notation do not affect the hard-coded donation limit of \$10,000,000., so if "1e12" is inputted, a popup notifies the user that the value must be less than or equal to \$10,000,000.

Issue #367: Zipcode has no error for invalid input

Changes can be found in the zipcode-formfield branch

<https://github.com/CSCD01/donate-wagtail-team34/tree/zipcode-formfield>

To view all changes in code, see the comparison here

<https://github.com/CSCD01/donate-wagtail-team34/commit/980b80921dd1fe2f4137460e1095912c67143511>

The ZIP code field raises a validation error now when it doesn't match the correct zip/postal code format. Before there were no checks on the format of the Zipcode field. Based on the country selected, the form field raises a ValidationError if it doesn't match the regex format.

This was implemented by adding a `clean()` method in `donate/payments/forms.py` to the `BraintreeCardPaymentForm` class in `donate/payments/forms.py` which inherits from Django's `forms.Form`. When the `clean()` method is called, `self.clean_data` will be populated with data that has survived so far. Here we get the data from the `country` and `post_code` field and using our dictionary of regexes in `donate/payments/constants.py` we raise a validation error if it doesn't match.

Issue #691: Add "no value selected" error state on the landing page

Changes can be found in the `dev/no_value_visa` branch

https://github.com/CSCD01/donate-wagtail-team34/tree/dev/no_value_visa

To view all changes in code, see the comparison here

<https://github.com/CSCD01/donate-wagtail-team34/commit/4cc851bd0f221ddf63b734b40793d9acecfbd3cd>

The bug fix prevents the landing page from refreshing when no donation amount is selected or inputted. Previously, the landing page refreshed when the user did not select/input a donation amount instead of showing a proper error message saying that no amount was selected.

The idea of implementation is to add a checking function in the frontend (i.e. `source/main.js`) which checks if the donation amount is 0. We are using the `FormData` to check the donation amount when the event 'submit' is triggered for both the `monthly-form` and the `single-form`. The variable `donateAmount` is initially 0 and when the event 'submit' is triggered we then use the `FormData` to update the `donateAmount`. If the `donateAmount` is still 0 during the submission we prevent the page from refresh and display a proper error message saying that no amount selected.

Issue #761: Clicking the Paypal Button triggers error if no amount is selected

Changes can be found in the `dev/no_value_paypal` branch

https://github.com/CSCD01/donate-wagtail-team34/tree/dev/no_value_paypal

To view all changes in code, see the comparison here

<https://github.com/CSCD01/donate-wagtail-team34/commit/47b1e6e2999096f09dbf514b94518a37e346656a>
<https://github.com/CSCD01/donate-wagtail-team34/commit/01583c081ad017d88cb33ea84627d54a526b02c9>

The bug fix now properly shows an error message to users who have clicked the paypal button to donate but not yet selected an amount to donate for. Previously, the landing page would show an error message saying "There was an error processing your payment. Please try again." instead of something helpful for the user to know what really went wrong.

The idea of implementation was to check which part of the component controls the error message when the user clicks paypal with no amount selected to donate for. After investigating through the frontend files within the architecture, we have figured out that they intended to add a Div html on the live server and most recent version of code that begins with style None and goes to Block whenever there is no amount being selected to block the paypal button from being clickable by the user. Now we have changed the fix to adding the error message to show up in that Div's onClick eventListener to address the issue. We have also contacted the admin on the donate-wagtail repo under this issue.

<https://github.com/mozilla/donate-wagtail/issues/761>



cadecairos commented on Jan 20

Member



STR:

1. Load the donation form
2. Click the Paypal button

Expected: An error about selecting an amount

Actual: A generic "There was an error processing your payment. Please try again." message appears.



cadecairos added the **engineering** label 7 hours ago



tagniam commented now



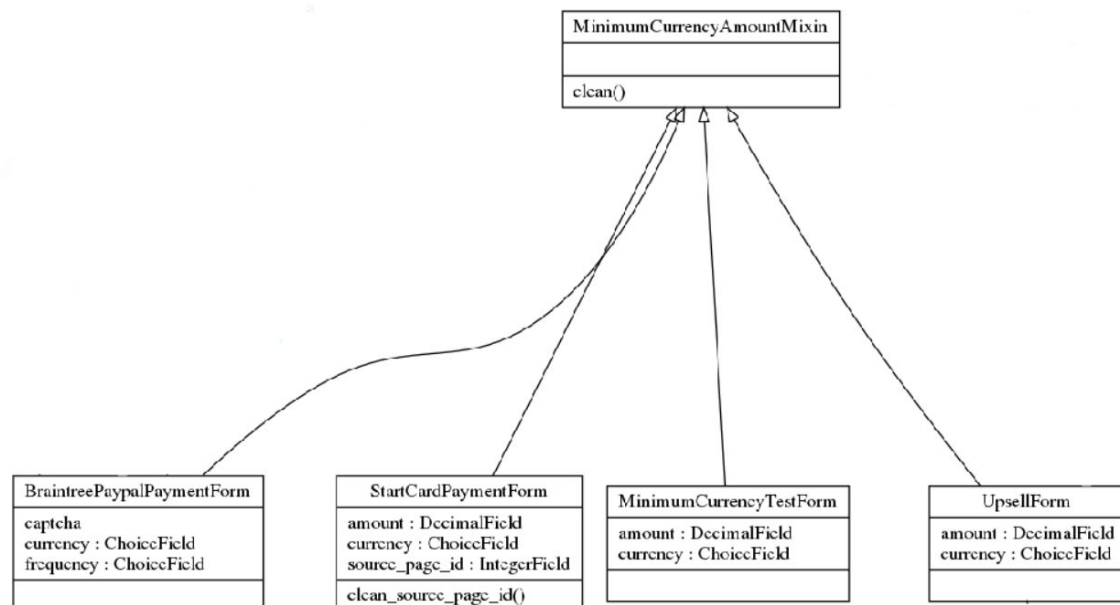
Hi @cadecairos, since this issue was opened, there's been a code change that prevents an error message from occurring at all.

It looks like an overlay (`payments__button--paypal--overlay`) on the PayPal button is inserted if there is no amount/an invalid amount in the Other Amount input (`amount-other-input`) to prevent the user from clicking the PayPal button: <https://github.com/CSCD01/donate-wagtail-team34/blob/master/source/js/payments-paypal.js#L106>

To fix this (i.e. actually display an error message) it looks like an error needs to be thrown in the on click listener when the input is invalid.

Design

Issue #661: "Other amount" text field allows amounts in scientific notation



We have modified the code in the `MinimumCurrencyAmountMixin`, and have tested each of its dependencies (`BraintreePaypalPaymentForm`, `StartCardPaymentForm`, and `UpsellForm`) through the unit testing suite. The `MinimumCurrencyTestForm` also runs correctly after the code modification, which passes our regression test.

Issue #367: Zipcode has no error for invalid input

Below is the updated UML diagram for the `BraintreeCardPaymentForm` class which now validates `post_code` in the `clean` method.



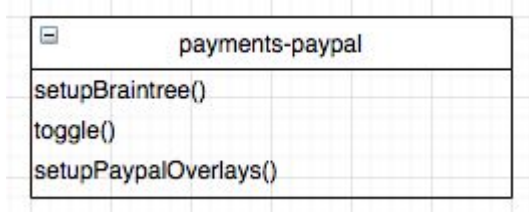
Issue #691: Add "no value selected" error state on the landing page

For issue 691, we only made some modifications to the frontend, more specifically the UI part (i.e. adding an error message), which did not change or affect the original design. So there is no point in adding a UML diagram here. We modified the `DOMContentLoaded` event listener function in the `main.js`. Lines 37-76 are the fix for the issue. We also added some

tests for the UI part to check the visibility of the error message using Selenium and Python. The landing page is displaying a proper error message when no amount is selected after the fix.

Issue #761: Clicking the Paypal Button triggers error if no amount is selected

For issue #761, the issue is only around the js files that control the frontend of the website. The most important change that we decided to make was adding the error message to show up every time user clicks on the invisible Div block that blocks the paypal button from being clickable by the user when there are no amount being selected rather than modifying code in the paypal.js since the paypal button is not clickable thus the error message will not be triggered by user clicking on the paypal button when there are no amount being selected to donate for. Below is the UML representation of payments-paypal.js and the issue was fixed by adding to show the error message in setupPaypalOverlays().



Acceptance Tests

In order to run the acceptance tests for the issues, make sure the target site is up and running properly. Instruction can be found here:

Follow instructions on <https://github.com/mozilla/donate-wagtail> to set up a local dev environment.

Issue #661: "Other amount" text field allows amounts in scientific notation

Acceptance test for one-time payments:

1. Visit the URL `http://0.0.0.0:8000/`.
2. Click the "One-Time" radio button to attempt a one-time donation.
3. Type "1e3" in the custom amount input box (the input box prefixed with a "\$" sign).
4. The input box should be cleared when the 'e' key is pressed, to prevent the user from entering numbers in scientific notation.

Acceptance test for monthly payments:

1. Visit the URL `http://0.0.0.0:8000/`.
2. Click the "Monthly" radio button to attempt a one-time donation.
3. Type "1e3" in the custom amount input box (the input box prefixed with a "\$" sign).
4. The input box should be cleared when the 'e' key is pressed, to prevent the user from entering numbers in scientific notation.

Issue #367: Zipcode has no error for invalid input

Acceptance tests can't be done in the docker environment because the error is checked once you submit the form to the backend but in the local docker environment, you can't submit the form. Refer to the unit tests below.

Issue #691: Add "no value selected" error state on the landing page

Before running the acceptance tests, go to `source/js/components/paypal.js` and comment out lines 40-43 and 51-54 since the paypal is disabled in the dev environment, there is an error message in the landing page. We need to comment out the lines in order to test our fix.

Acceptance test for one-time payments:

1. Visit the URL <http://0.0.0.0:8000/>.
2. Click the "One-Time" radio button
3. Click on the "visa" button under the "Choose Payment" section
4. The page should not refresh and the error message should be displayed
5. The error message should be: "No amount selected! Please select a donation amount!"

Acceptance test for monthly payments:

1. Visit the URL <http://0.0.0.0:8000/>.
2. Click the "Monthly" radio button
3. Click on the "visa" button under the "Choose Payment" section
4. The page should not refresh and the error message should be displayed
5. The error message should be: "No amount selected! Please select a donation amount!"

Issue #761: Clicking the Paypal Button triggers error if no amount is selected

Acceptance test:

- 1: Comment out lines 40-43 and 51-54 in `/src/js/components/paypal.js` to have the paypal button show up in the dev version.
- 2: Open up the browser and go to <http://localhost:8000/>.
- 3: Click the PayPal button.
- 4: An error at the top should occur with the message "You have not selected an amount to donate for. Please try again." rather than "There was an error processing your payment. Please try again.".

Unit Test Suite

Issue #661: "Other amount" text field allows amounts in scientific notation

There are also unit tests written in the `donate/core/tests/test_models.py` file, that tests the input sanitization on the backend. The unit tests can be run with the ``inv docker-test-python`` command. The unit test for this bug attempts to make a donation with the amount `'1e1000'` in dollars, which is prevented by setting the donation amount to 0 as a precaution instead.

Issue #367: Zipcode has no error for invalid input

There are also unit tests in `donate/payments/tests/test_forms.py` file which also tests the input sanitization for the payment form. All the tests can be run with ``inv docker-test-python`` command. There are unit tests for valid and invalid US and Canadian zip/postal codes to ensure it is checking the format of the input properly.

Issue #691: Add "no value selected" error state on the landing page

Since we are testing the visibility of the DOM elements (i.e. the error message), we decided to go with Selenium and Python for the unit tests since there are no tests for the UI part in the original design.

Test cases are written in the folder **donate/payment/tests**, we created 4 test cases, one-time, monthly, and one-time, monthly in a different currency (ie French currency). Code can be found below

https://github.com/CSCD01/donate-wagtail-team34/tree/dev/no_value_visa/donate/payments/tests

- **test_no_value_monthly_fr.py**
- **test_no_value_onetime_fr.py**
- **test_no_value_visa_monthly.py**
- **test_no_value_visa_onetime.py**

How to run:

1. Open terminal and run `pip install -U selenium` to install the python binding
2. `cd` to the directory `../donate-wagtail-team34/donate/payments/tests`.
3. run `python test_no_value_monthly_fr.py` to see if the error message displayed on the French currency landing page with the Monthly panel.
4. run `test_no_value_onetime_fr.py` to see if the error message displayed on the French currency landing page with the One-time panel.
5. run `test_no_value_visa_monthly.py` to see if the error message displayed on the default landing page with the Monthly panel.
6. run `test_no_value_visa_onetime.py` to see if the error message displayed on the default landing page with the One-time panel.
7. verify and validate the results.

Issue #761: Clicking the Paypal Button triggers error if no amount is selected

We have decided to use Selenium's web driver with Python for the unit tests for this since it is only testing for certain UI element's value on the page and not something backend focused. Test cases are in the folder **donate/payment/tests**, we created 4 test cases. Code can be found below

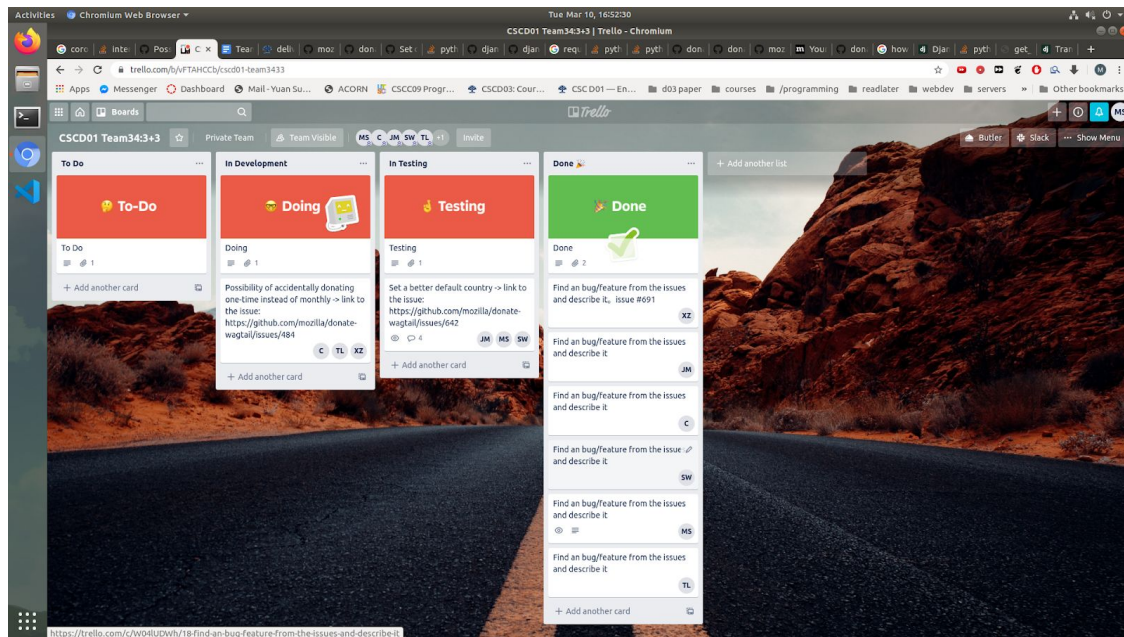
https://github.com/CSCD01/donate-wagtail-team34/tree/dev/no_value_paypal/donate/payments/tests

- **test_no_value_paypal_monthly_fr.py**
- **test_no_value_paypal_onetime_fr.py**
- **test_no_value_paypal_monthly.py**
- **test_no_value_paypal_onetime.py**

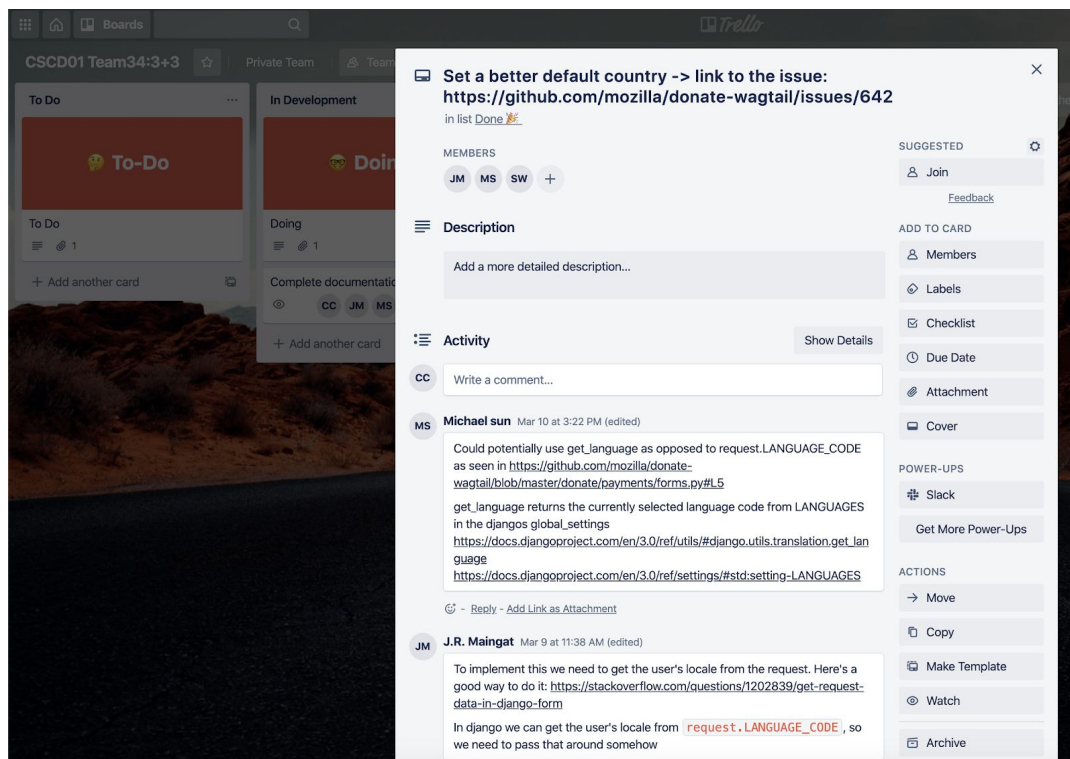
How to run:

1. Open terminal and run `pip install -U selenium` to install the python binding
2. `cd` to the directory `../donate-wagtail-team34/donate/payments/tests`.
3. run `python test_no_value_paypal_monthly_fr.py` to see if the error message displayed on the French currency landing page with the Monthly radio selected but no amount selected.
4. run `test_no_value_paypal_onetime_fr.py` to see if the error message displayed on the French currency landing page with the One-time radio and no amount selected
5. run `test_no_value_paypal_monthly.py` to see if the error message displayed on the default landing page with the Monthly radio but no amount selected.
6. run `test_no_value_paypal_onetime.py` to see if the error message displayed on the default landing page with the One-time radio but no amount selected.
7. verify and validate the results.

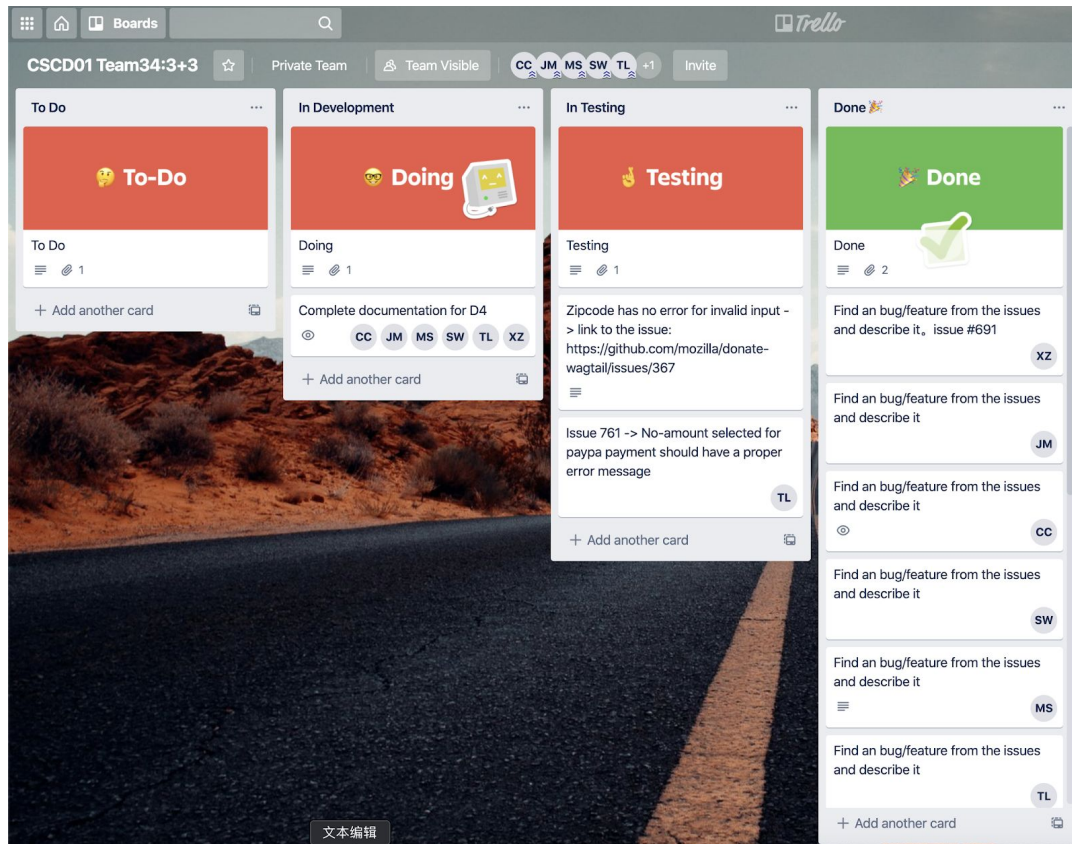
Software development process



screenshot of the kanban board during D2



screenshot of one of the cards on the kanban board in D2



screenshot of the kanban board for deliverable4

We strictly follow the Kanban development process from Deliverable 1 to Deliverable 4. For this deliverable, We created 4 cards for each major bug, each card includes the information(name, description, detail, status) about the task and team members who work on it. And 1 card for the report of deliverable 4.

Each card goes through the workflow due to the completeness.

In the TO-DO stage, we design the architecture of the specific bug, including drawing the UML diagram, reviewing code and communicating with Mozilla employees through Github.

In the In-Development stage, we work on the implementation of each bug and writing test suite.

In the In-Testing stage, we test our implementation and modify it based on the test results.

After all test cases have been passed successfully, we can move the card to the Done stage.

Our group holds an online meeting through **discord** every week, team members report on their work progress and discuss their goal for the next stage. The total number of cards that are working at once are strictly never surpassing the WIP limits(6).